

# python语言程序设计

## Design and Programming of Python Language



主讲教师：张小东

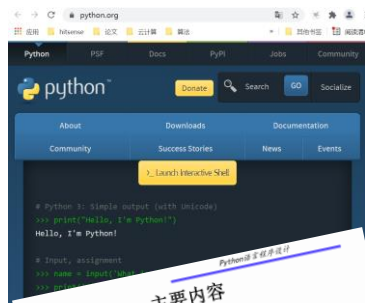


方式：[z\\_xiaodong7134@163.com](mailto:z_xiaodong7134@163.com)

地点：哈尔滨工业大学(威海)  
研学楼422室

群 号:693864916  
该群暂时不能被搜索  
请设置群头像和群名称

## ◆ 参考资料



<https://www.python.org/>

教python那个教师给的学习资料



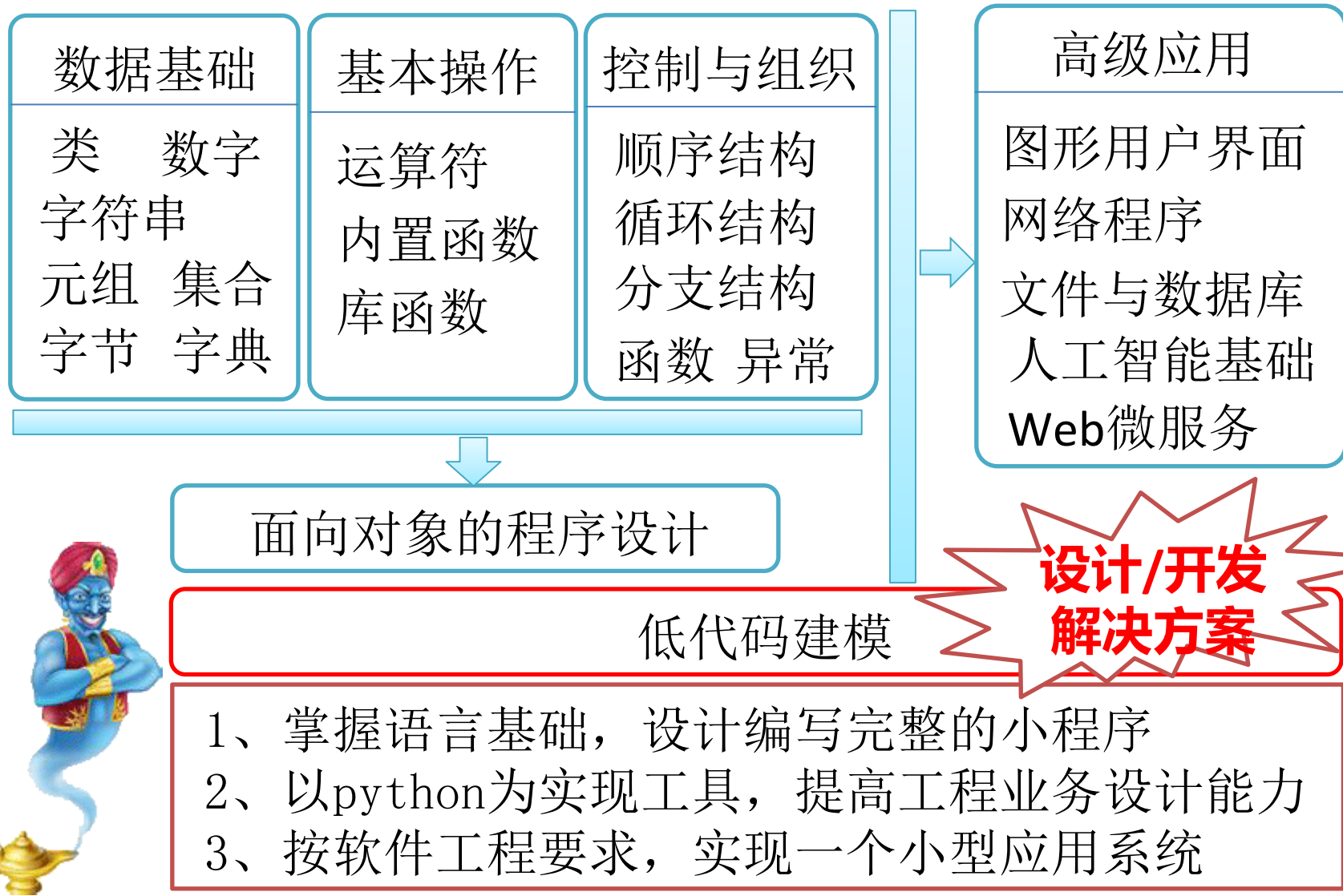
《python基础教程》，Magnus Lie Hetland, 人民邮电出版社

- 课时安排  
— 授课：32学时

**成绩构成：**

**平时作业10% + 综合作业20% + 翻转课堂10%  
+ 期末考试60%**

# 课程体系结构



# 第1章 python语言编程入门

## 主要内容

- **python**简介
- **python**的基本语法
- 类与对象的传说
- 基本数据类型及操作
- python库的使用

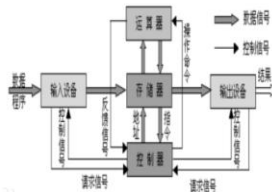
===python的基本语法===

? 你认为是什么推动了计算机技术的发展?



ENIAC

约翰·冯·诺依曼



1. 计算机系统结构阶段



2. 计算机网络和视窗



3. 大数据、互联网+, 云计算...



4. ChatGPT、DeepSeek...

? 请问你的Pyhton学习目标是什么?

## ===python简介===



Guido van Rossum

- ◆ python 20世纪80年代末由荷兰人**Guido van Rossum**设计实现的。
- ◆ 1991年，**Ver 0.9.0 python**，实现了类、函数以及列表、字典和字符串等基本的数据类型，集成了模块系统。
- ◆ 1994年，**python 1.0**发布了。**1.0**新增了函数式工具。

当前最高版本：python3.13.2

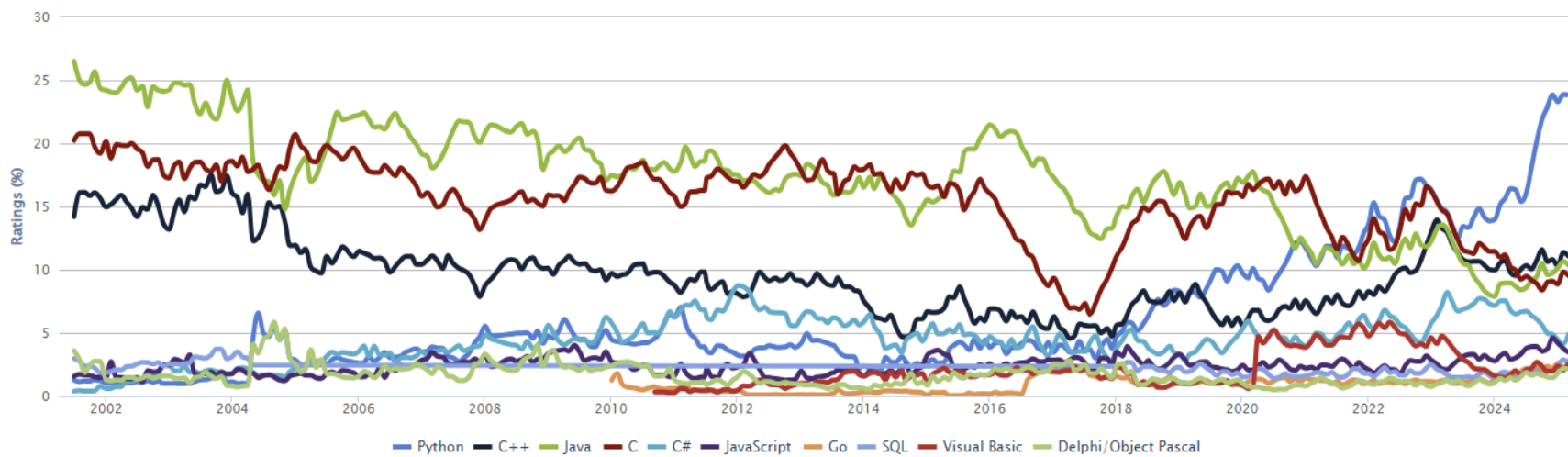
- ◆ 2001年，推出**python 2.0**，它集成了列表推导式(List comprehension)。**优雅 明确 简单**
- ◆ 2009年，推出**python 3.0**，为了不带入过多的累赘，**python 3.0**在设计的时候没有考虑向下兼容。

## ===python简介===

Mar 2025	Mar 2024	Change	Programming Language		Ratings	Change
1	1			Python	23.85%	+8.22%
2	3	▲		C++	11.08%	+0.37%
3	4	▲		Java	10.36%	+1.41%
4	2	▼		C	9.53%	-1.64%
5	5			C#	4.87%	-2.67%
6	6			JavaScript	3.46%	+0.08%

TIOBE Programming Community Index

Source: www.tiobe.com





## ===python简介===

◆ python的安装 <https://www.python.org/downloads/>

python 3.13.2

Python version	Maintenance status	First released	End of support	Release schedule
3.14	pre-release	2025-10-01 (planned)	2030-10	PEP 745
3.13	bugfix	2024-10-07	2029-10	PEP 719
3.12	bugfix	2023-10-02	2028-10	PEP 693
3.11	security	2022-10-24	2027-10	PEP 664

## 集成开发工具

- 1.vscode+python extension for VSCode
- 2.JetBrains PyCharm

新特性、改进现有  
功能或提供信息的  
文档



# 第1章 python语言编程入门

## 主要内容

- **python**简介
- **python**的基本语法
- 类与对象的传说
- 基本数据类型及操作
- python库的使用

## ===python的基本语法===

## 2.1两种运行模式：命令行交互与程序文件

```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> name = input("输入姓名: ")
输入姓名: 郭靖
>>> print("%s同学, 学好python, 好找工作! "%name)
郭靖同学, 学好python, 好找工作!
>>> print("{}大侠, 学好python, 功夫与思维均长!".format(name[0]))
郭大侠, 学好python, 功夫与思维均长!
>>> print("{}哥哥, 学好python, 提升魅力值!".format(name[1:]))
靖哥哥, 学好python, 提升魅力值!

```

IDLE 命令行交互

对象方法调用

```

VSCode
文件(F) 编辑(E) 选择(S) 查看(V) ... python
资源管理器 1 个未保存
打开的编辑器
first.py
设置 settings.json
PYTHON
first.py
hello.py
first.py
1 name = input("输入姓名: ")
2 print("%s同学, 学好python, 好找工作! "%name)
3 print("{}大侠, 学好python, 功夫与思维均长!".format(name[0]))
4 print("{}哥哥, 学好python, 提升魅力值!".format(name[1:]))
5
问题 输出 调试控制台 终端 端口
PS D:\vscode\python> & C:/Users/z_xia/AppData/Local/Microsoft/WindowsApps/python3.12.exe d:/vscode/python/first.py
输入姓名: 郭靖
郭靖同学, 学好python, 好找工作!
郭大侠, 学好python, 功夫与思维均长!
靖哥哥, 学好python, 提升魅力值!

```

VSCode 中以文件形式运行

# ===python的基本语法=== 2.2 语句与段落

? 还记得C语言中的小兔子吗?

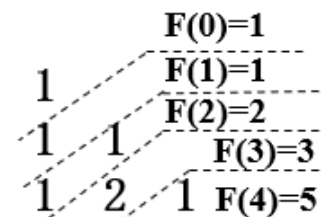


(意)

$n = 1$

$n = 2$

$n > 2$



$$a_i/a_{i-1}$$

那契数列

```
>>> a, b = 0, 1
>>> while a < 1000:
    print(a, end = ', '); a, b = b, a + b
```

多条语句

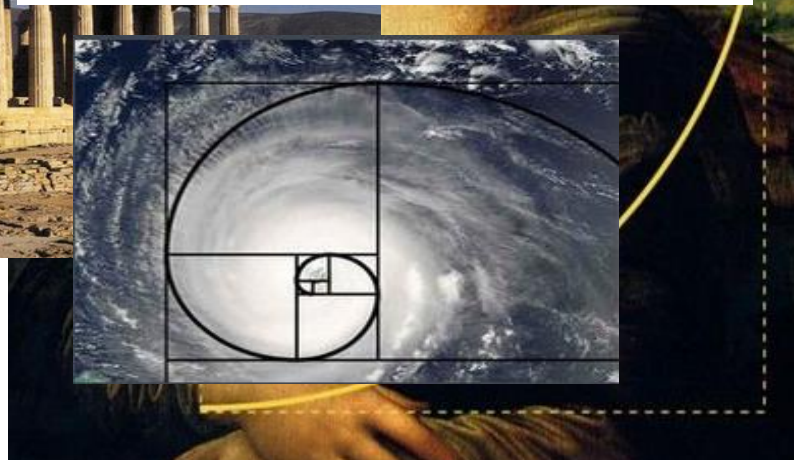
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987,



Baidu 百科

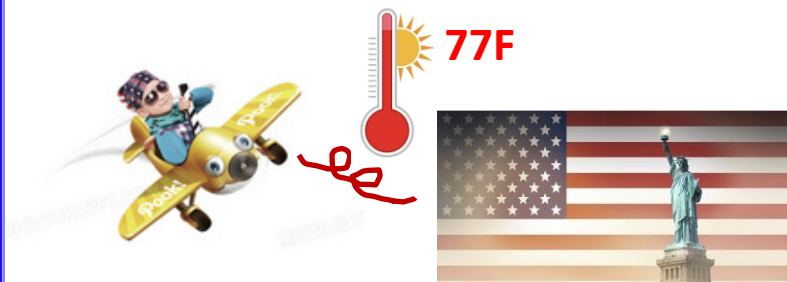


Baidu 百科



===python的基本语法===

## 2.3 python语法元素分析



【例题1-1】一个hitter开着飞机去美国看女神。到达纽约上空时，指挥塔告诉他：当前面温度**77F**。问转换为摄氏度为多少？

【问题分析】 1标准大气压下水的结冰点与沸点

摄氏度(Celsius)：0C，100C；华氏度(Fahrenheit)：32F, 212F

转换公式： $C = (F - 32) / 1.8$  ；  $F = C * 1.8 + 32$

【IPO原则】

输入：带华氏或摄氏标志的温度值

处理：根据温度标志选择适当的温度转换算法

输出：带摄氏或华氏标志的温度值

===python的基本语法===

## 2.3 python语法元素分析

【例题1-1】当前面温度77F时...



CFSwitch.py &gt; ...

```

1  # Both of temperatures Convert each other
2  '''
3  author: zxd
4  date:2025-03-05
5  '''
6  TempStr=input("请输入带有所号的温度值: ")
7  if TempStr[-1] in ['F','f']:
8      C = (eval(TempStr[0:-1]) - 32)/1.8
9      print("转换后的温度是{:.2f}C".format(C))
10 elif TempStr[-1] in ['C','c']:
11     F = 1.8*eval(TempStr[0:-1]) + 32
12     print("转换后的温度是{:.2f}F".format(F))
13 else:
14     print("输入格式错误")
15

```

PEP 572

PEP 8

下列语法元素出现在程序中的哪些地方 ?

1. 注释
2. 标识符
3. 关键字
4. 表达式
4. 函数/预定义标识符
5. 语句
6. 段落



技巧

&gt;&gt;&gt;help()

Help&gt;keywords

Help&gt;return

问题 输出 调试控制台 终端 端口

PS D:\vscode\python> & C:/Users/z\_xia/AppData/Local/Mi  
 请输入带有所号的温度值: 77f  
 转换后的温度是 25.00C

其实挺好的!



Beller me!



===python的基本语法===

# 小结



请问python架构风格是什么？



本小节当中你学到哪些语法元素？



试编写一个程序完成角度与弧度转换？

# 第1章 python语言编程入门

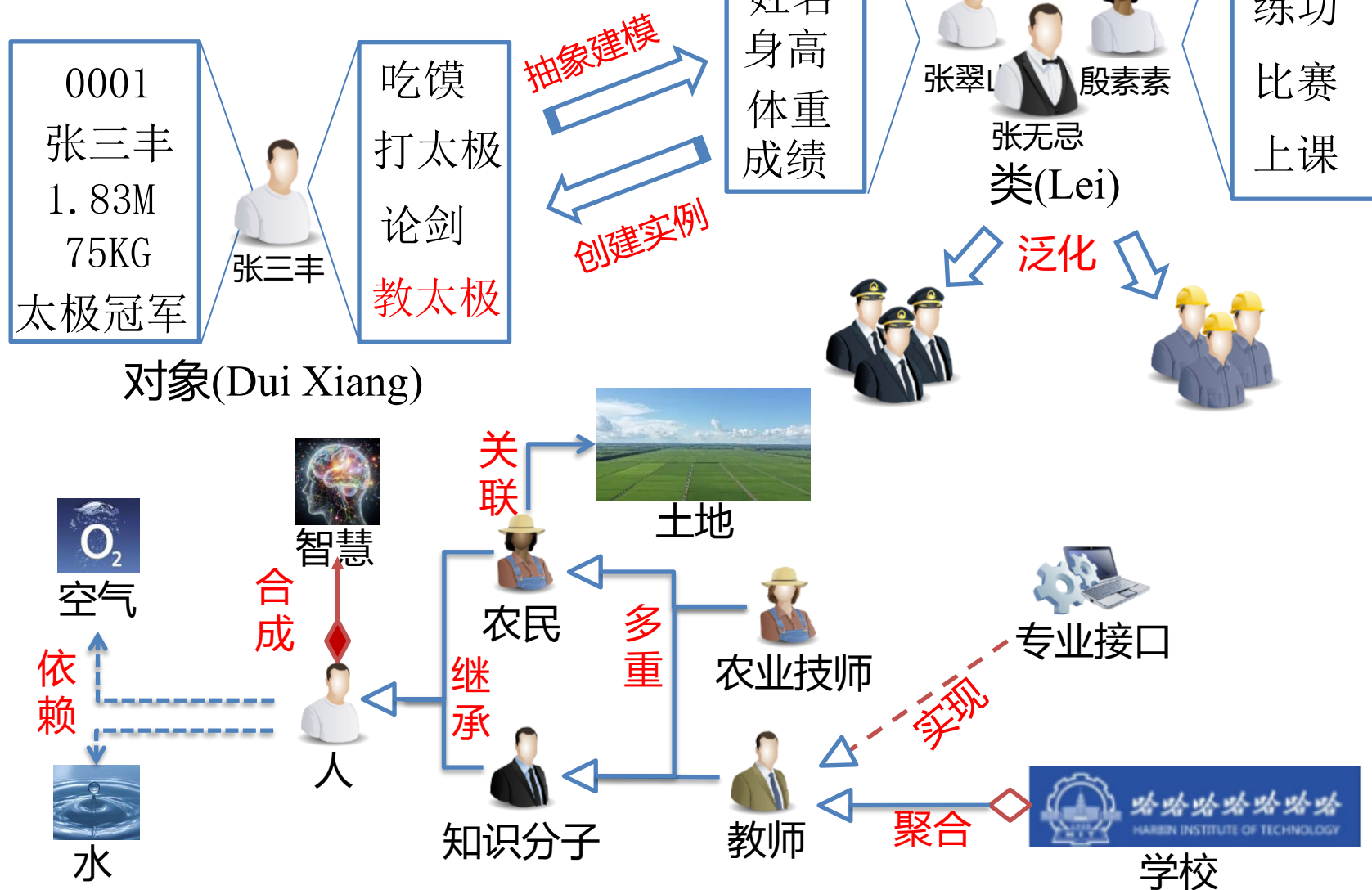
## 主要内容

- **python**简介
- **python**的基本语法
- 类与对象的传说
- 基本数据类型及操作
- python库的使用



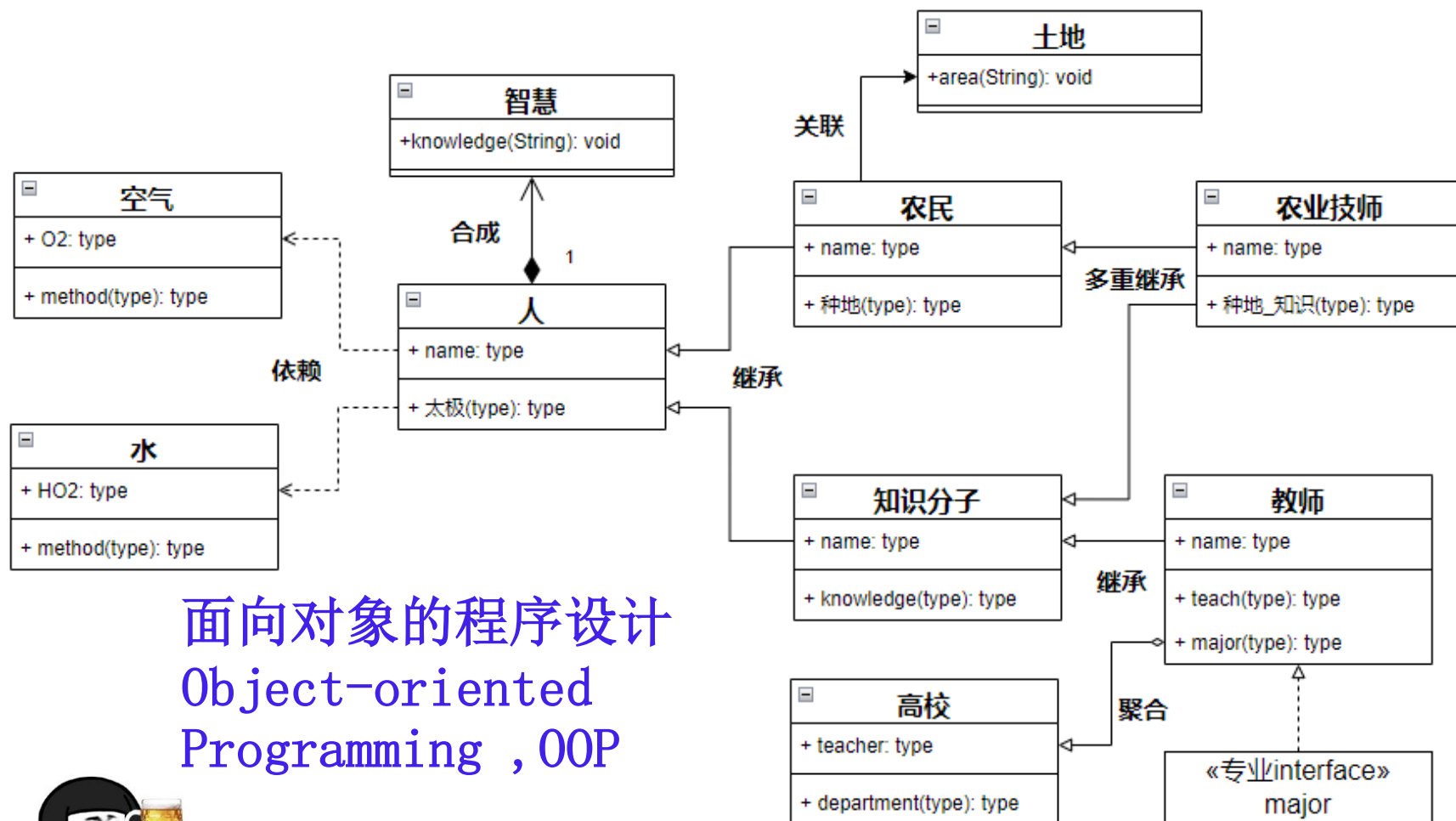
## ===类与对象===

## 3.1 对象(Object)与类(Class)



## ===类与对象===

## 3.2 类图—类之间的逻辑关系



面向对象的程序设计  
Object-oriented  
Programming , OOP



python — 一切皆对象！

## ===类与对象===

## 3.3 python中对象与类定义与使用

```
>>> name = “张三丰” ⇐ # 1. 声明字符串对象
```

```
>>> name.__len__() ⇐ # 2. 对象的方法调用
```

```
class ClassName(Base1, Base2, Base3): ⇐ # 3. 定义类
```

```
# 定义私有属性, 在类外部无法直接进行访问
```

```
# 定义构造方法
```

```
def __init__(self,...):
```

```
...
```

```
# 定义其它方法
```

```
...
```

```
# 定义析构方法
```

```
def __del__(self,...):
```

```
...
```

## #4. 观测对象

**id():** 获取对象的标识

**type():** 获取对象类型

>>> id(11)	>>> type(12)
1657845120	<class 'int'>
>>> id('python')	>>> type(1.2)
48172648	<class 'float'>

类规定了对象所具有的属性和操作这些属性的方法

===python的基本语法===

# 小结

 请问类与对象是什么关系？

 类间关系有几种？ 这些关系的作用是什么？

# 第1章 python语言编程入门

## 主要内容

- **python**简介
- **python**的基本语法
- 类与对象的传说
- 基本数据类型及操作
- python库的使用

## ===基本数据类型及操作===

## 4.1 变量与常量

变量：以赋值(=)方式来声明变量(对象)

```
>>> a=112
```

```
>>> type(a)
```

```
<class 'int'>
```

```
>>> id(a)
```

```
1657848352
```

```
>>> id(112)
```

```
1657848352
```

```
>>> b=112
```

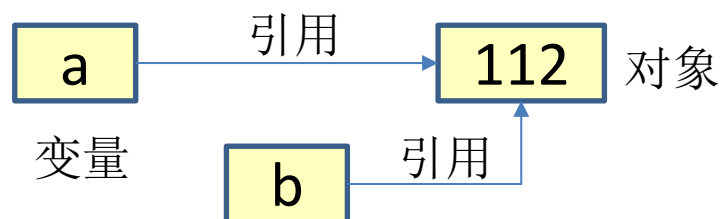
```
>>> id(b)
```

```
1657848352
```

创建一个变量a

a指向整数类型对象

id标识一样



PEP

```
>>> x, y, z = 5, 7, 8
```

```
>>> a = 5
```

```
>>> a.bit_length()
```



技巧 Python没有提供内置的方式来声明**常量**（象C中用const修饰）  
可以用约定的形式用变量来模拟或用相关类库

## ===基本数据类型及操作===

## 如何查看数据类型 ?

## 4.2 数字类型及其操作

【分类】整数、浮点数(科学计数法)、高精度数、复数、分数

&gt;&gt;&gt; import decimal #1.高精度数

&gt;&gt;&gt; a = decimal.Decimal('3.141592653')

&gt;&gt;&gt; b = decimal.Decimal('1.234567898')

&gt;&gt;&gt; decimal.getcontext().prec=20

&gt;&gt;&gt; a\*b

Decimal('3.878509437986453394')

&gt;&gt;&gt; x = 1.25

&gt;&gt;&gt; a = 1.2e-3

&gt;&gt;&gt; b = 12.3 + 1j

&gt;&gt;&gt; from fractions import Fraction #2.分数

&gt;&gt;&gt; Fraction(12,20) #Fraction(3, 5)

&gt;&gt;&gt; Fraction(1.25) #Fraction(5, 4)

&gt;&gt;&gt; a=Fraction(1,3)

&gt;&gt;&gt; b=Fraction(2,3)

&gt;&gt;&gt; print(a+b) #1

a=Fraction(1.6)

print(a)

输出: ?



## ===基本数据类型及操作===

## 4.2 数字类型及其操作

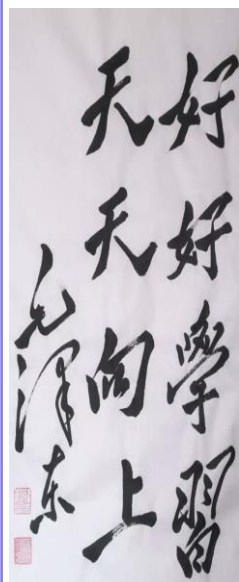
## 【操作】操作符、内置函数、类方法

(1) 操作符: +, -, \*, /, %, //(整除), \*\*(幂运算), 二元操作符+=...

(2) 内置函数: abs(x), divmod(x, y), pow(x, y [,z]),  
round(x[, ndigits]), max(a, b, c,...), min(a, b, c,...)

类型转换: int(x), float(x), complex(re[,im])

【例题1-2】请用每天努力的效率诠释毛泽东的提词寄语?



```
ex=float(input("输入每天的提升效率:"))
dayup = (1.0 + ex)**365
daydown = (1.0 - ex)**365
print("向上: {:.2f}, 向下: {:.2f}.".format(dayup, daydown))
```

输入每天的提升效率:0.01  
向上: 37.78, 向下: 0.03.



## === 基本数据类型及操作 ===

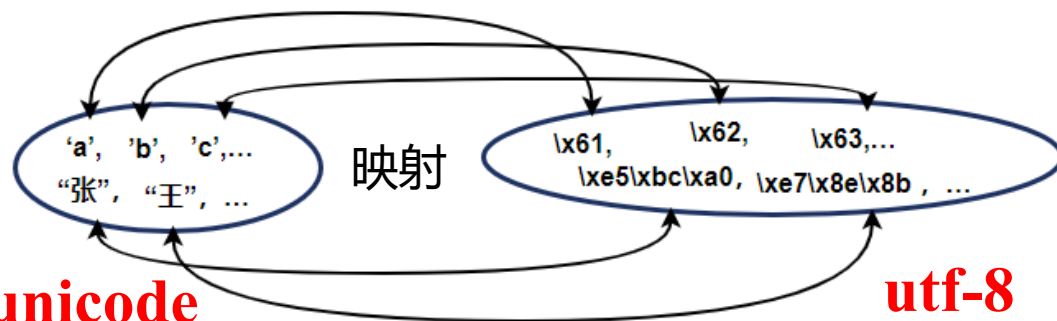
## 4.3 字符串及其操作

## 【字符集与编码方式】

1. **Unicode**: 全球性的 **unicode**

字符集，为世界上所有字符分配唯一的编号(称为码位或码点)，实现**跨语言**、**跨平台**的文本处理，它解决了不同文字系统间**传递**文本时出现的**乱码**问题。

2. **UTF-8**: Unicode的一种实现方式，使用变长的编码规则，将Unicode的码位转换为字节序列，使用1到4个字节表示一个字符，优化了存储空间并兼容ASCII码。



## 【声明字符串变量】

```
>>> a = '单引号'
```

```
>>> b = "双引号"
```

```
>>> c = """三引号嵌套"双引号"和'单引号',可以换行"""
```

```
>>> a
'单引号'
>>> b
"双引号"
>>> c
"""三引号嵌套"双引号"和\'单引号\',可以换行"""
```

## ===基本数据类型及操作===

## 4.3 字符串及其操作

## 【操作】

1. 支持正向和反向索引
2. 支持片选: [N:M],[[:M],[N:]

如: `hi = "hello,hitter"; hi[0:5] # 'hello'; hi[-1] # 'r'`

`hi[0:-1] # 'hello,hitte'`

3. 操作符: `x + y`, `x*n`, `x in s`, `x not in s`

【例题1-3】获取星期字符串。程序读入一个表示星期儿的数字(1~7), 输出对应的星期字符串名称。

```
weekstr = "星期一星期二星期三星期四星期五星期六星期日"
weekid = int(input("请输入星期数字(1-7): "))
pos = (weekid - 1)*3
print(weekstr[pos: pos+3])
```



索引

反向递减序号

-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
h	e	l	l	o	,	h	i	t	t	i	e	r
0	1	2	3	4	5	6	7	8	9	10	11	12

正向递增序号



## === 基本数据类型及操作 ===

## 4.3 字符串及其操作

## 【操作】4. 内置函数（6个）

函数	功能描述
len(x)	返回字符串x的长度。 也可返回其他组合数据类型元素个数
str(x)	返回任意类型x所对应的字符串形式
chr(x)	返回Unicode编码x对应的单字符
ord(x)	返回单字符表示的Unicode编码
hex(x)	返回整数x对应十六进制数的小写形式字符串
oct(x)	返回整数x对应八进制数的小写形式字符串

## 【操作】5. 字符串对象函数（16个）

函数	功能描述
str.lower()	返回字符串str的副本，全部字符小写
str.upper()	返回字符串str的副本，全部字符大写
str.islower()	当str所有字符都是小写时，返回True,否则返回False

## === 基本数据类型及操作 ===

## 4.3 字符串及其操作

## 【操作】5. 字符串对象函数（16个）

函数	功能描述
str.split(sep=None, maxsplit=-1)	返回一个列表，由str根据sep被分隔的部分构成
str.replace(old new[, count])	返回字符串str的副本，所有old子串被替换为new,如果count给出，则前count次old出现被替换
str.strip([chars])	返回字符串str的副本，在其左侧和右侧去掉chars中列出的字符
str.zfill(width)	返回字符串str的副本，长度为width,不足部分在左侧添0
str.count(sub[, start[, end]])	返回str[start, end]中sub子串出现的次数
str.format()	返回字符串str的一种排版格式
str.join(iterable)	返回一个新字符串，由组合数据类型iterable变量的每个元素组成，元素间用str分隔
.....	.....

## === 基本数据类型及操作 ===

## 4.3 字符串及其操作 【操作】 5. 字符串对象函数 (16个)

## format()方法的基本使用

(1) 格式：&lt;模板字符串&gt;.format(&lt;逗号分隔的参数&gt;)

**槽**

```
>>> "{:}:计算机{:}的CPU占用率为{:}%".format("2025-4-3","PYTHON",10)
'2025-4-3:计算机PYTHON的CPU占用率为10%.'
```

槽:    0                    1                    2                    参数:    0                    1                    2

```
>>> "{1}:计算机{0}的CPU占用率为{2}%".format("2025-4-3","PYTHON",10)
'PYTHON:计算机2025-4-3的CPU占用率为10%.'
```

## ===基本数据类型及操作===

## 4.3 字符串及其操作【操作】 5. 字符串对象函数（16个）

## format()方法的基本使用

(2)槽的内部样式如下：{&lt;参数序号&gt;:&lt;格式控制标记&gt;}

:	<填充>	<对齐>	<宽度>	<,>	<精度>	<类型>
引导符号	用于填充的单个字符	< 左对齐 > 右对齐 ^ 居中对齐	槽的设定输出宽度	数字的千位分隔符,适用于整数和浮点数	浮点数小数部分的精度或字符串的最大输出长度	整数类型: b,c,d,o,x,X 浮点数类型: e,E,f,%

&gt;&gt;&gt; s="python"

&gt;&gt;&gt; n=123456.6789

&gt;&gt;&gt; "{1:\*&gt;10}:{0:-^20,.2f}".format(n,s)

输出: '\*\*\*python:-----123,456.68-----'



# 第1章 python语言编程入门

## 主要内容

- **python**简介
- **python**的基本语法
- 类与对象
- 基本数据类型及操作
- **python**的类库及使用

## ==python的类库及使用==

## 5.1 基本概念

【模块】具有相对完整功能的代码集合python文件,解决**代码复用**问题。

```
# math_utils.py 模块定义
def sqrtz(x):
    """计算 x 的开方"""
    return x **0.5
```

【包】管理 Python 模块命名空间的方式

↓ 引用并调用

```
# main.py
import math_utils # 引用模块
print(math_utils.sqrtz(5)) # 输出 2.236
```

```
1 data_processing/
2 |
3 |— __init__.py      # 标记为包
4 |— text_processing/
5 |   |— __init__.py  # 标记为包
6 |   |— text_utils.py
7 |— image_processing/
8 |   |— __init__.py  # 标记为包
9 |   |— image_utils.py
10 |— audio_processing/
11 |   |— __init__.py # 标记为包
12 |   |— audio_utils.py
```

```
from data_processing.text_processing import text_utils
from data_processing.image_processing import image_utils
```

## 5.1 基本概念

【库】成套的工具集合。pandas 为数据分析的库，包括(Series 和 DataFrame 等模块)。

```
import pandas as pd  
  
ser1 = pd.Series(data=[120, 2, 3, 4], index=['一季度', '二季度', '三季度', '四季度'])  
  
print(response.json())
```

【例题1-4】利用math库函数判断一个4位整数是否为回文数。

```
from math import *  
  
n = int(input("请输入一个4位整数:"))  
  
m = n//1000 + n//100%10*10 + n//10%10*100 + n%10*pow(10,3)  
  
print("数{}判断是否为回文数的结果为: {}".format(n, n == m))
```

## 5.2 python类库的使用

【math库】用于数学计算的标准函数库，提供4个数学常数和44个函数。

```
import math
math.factorial(10) #求阶乘
math.gamma(10) #?
```



### 技巧

$\int_0^{\infty} x^t e^{-x} dx$  伽玛函数（欧拉第二积分函数），伽玛函数的推广：

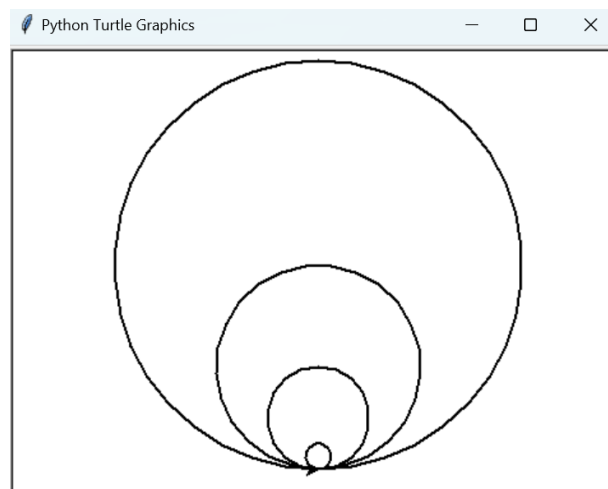
- (1)  $x$ 为任意数， $\Gamma(x+1) = x\Gamma(x)$
- (2) 当 $x$ 为整数时， $\Gamma(x+1) = x\Gamma(x) = x(x-1)\Gamma(x-1) = \cdots = x!$
- (3)  $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$

## ==python的类库及使用==

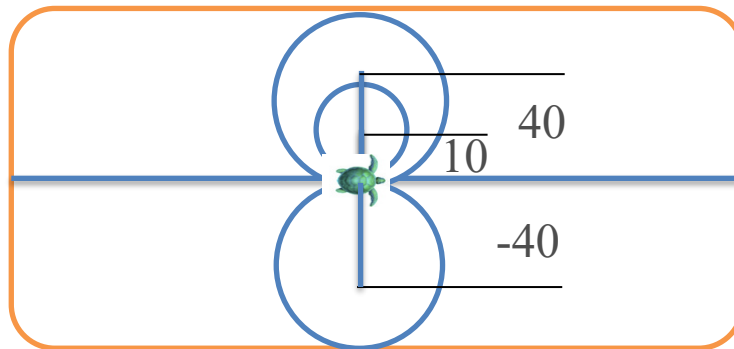
## 5.2 python类库的使用

【turtle库】 内置的绘图标准库，通过一组函数控制画笔的行进动作，完成绘制。

```
import turtle  
turtle.pensize(2) #画笔粗2个像素  
turtle.circle(10) #半径10个像素点  
turtle.circle(40)  
turtle.circle(80)  
turtle.circle(160)
```



小海龟在图的正中(0,0), 头向右, 尾向左。Circle参数为10, 小海龟身体左侧10为原点, 为-40负右侧40为原点。

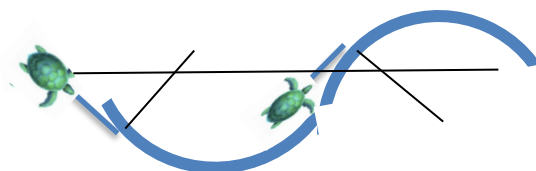


## 5.2 python类库的使用



turtle库绘制图形有一个基本框架:一个小海龟在坐标系中爬行,其爬行轨迹形成了绘制图形。对于小海龟来说,有“前进”、“后退”、“旋转”等爬行行为,对坐标系的探索也通过“前进方向”、“后退方向”、“左侧方向”和“右侧方向”等小海龟自身角度方位来完成。

```
seth(-40) #角度  
for i in range(4):  
    circle(40,80) # 半径, 角度  
    circle(-40, 80)
```



## ==python的类库及使用==

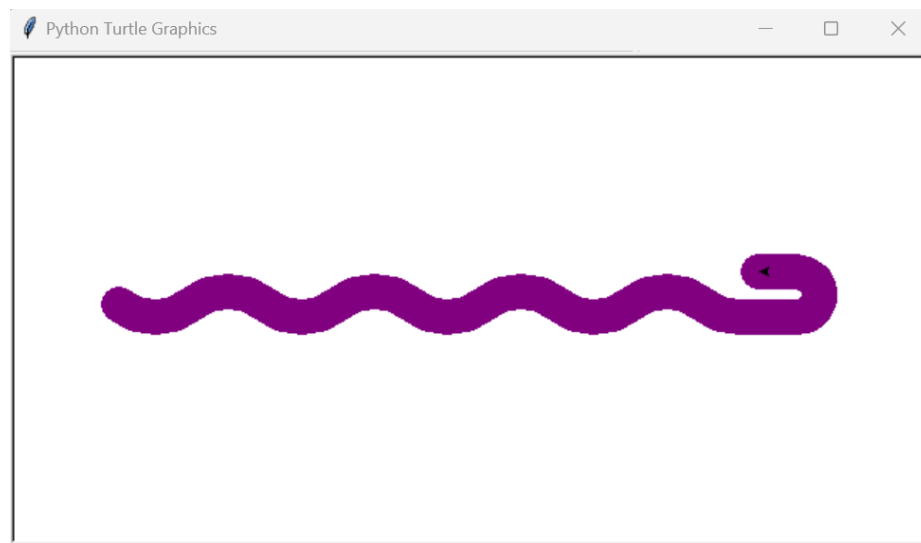
## 5.2 python类库的使用

【例题1-5】 利用turtle库画一条大python。

```

from turtle import *
setup(650,350,200,200) #主窗体大小及位置
penup() #抬笔
fd(-250) #倒退
pendown() #落笔
pensize(10) #笔线粗25
pencolor("purple") #颜色紫色
seth(-40) #角度
for i in range(4):
    circle(40,80) # 半径, 角度
    circle(-40, 80)
circle(40,80/2)
fd(40)
circle(16,180)
fd(40*2/3)

```





# 本章小结

- python的学习内容与计划
- python的基本概念
- python的基本语法